RESEARCH ARTICLE OPEN ACCESS

Job Scheduling in Big Data – A Survey

P.Divya*, M.Ramesh**

*(Department of CSE, CMS College of Engineering, and Namakkal
Email: divyapoomalai@gmail.com)

** (Department of CSE, CMS College of Engineering, and Namakkal
Email: mramesh9867@gmail.com)

--************************----------------------------------

Abstract:

To quantify the large amount of data stored electronically is not easy. The large amount of data (Data is in the unit of zettabytes or exabytes) is called Big Data. To process these large datasets Hadoop system is used. To collect these big data according to the request Map Reduce program is used. To achieve greater performance in process big data proper scheduling is required. To minimize starvation and maximize resource utilization, Scheduling is a technique of assigning jobs to available resources in a manner. Scheduling technique Performance can be improved by applying deadline constraints on jobs. The objective of the research is to study and analyze various scheduling algorithms for better performance.

Keywords **—Big Data, Map Reduce, Hadoop, Job Scheduling Algorithms.**

--************************----------------------------------

I. INTRODUCTION

Currently, the term big data [1] has become very trendy in Information Technology segment. Big data refers to broad range of datasets which are hard to be managed by previous conventional applications. Big data can be originate in finance and business, banking, online and onsite purchasing, healthcare, astronomy, oceanography, engineering, and many other fields. These datasets are very difficult and are rising exponentially day by day in very large amount. Data comes from social media sites, sensors, digital photos, business transactions etc. As data is increasing in volume, in variety and with high velocity, it guides to complexities in processing it. To correlate, link, match, and transform such big data is a complex process. Big data being a developing field has a lot of research problems and challenges to address. The major research problems in big data are following: 1) Handling data volume, 2) Analysis of big data, 3) Privacy of data, 4) Storage of huge amount of data, 5) Data visualization, 6) Job scheduling in big data, 7) Fault tolerance. 1)

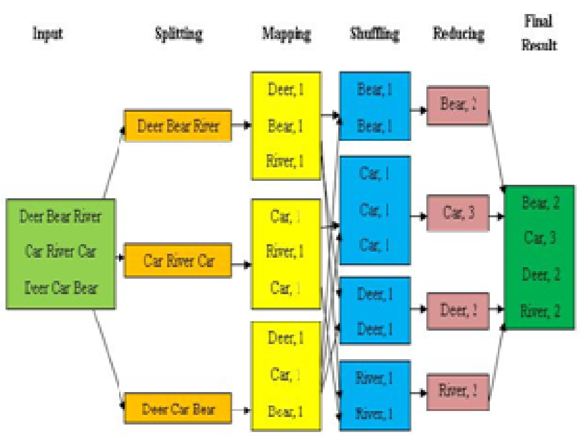
Handling data volume [1] [2]: The large amount of data coming from different fields of science such as biology, astronomy, meteorology, etc make its processing very difficult computing to scientists. 2) Analysis of big data: it is difficult to analyze big data due to heterogeneity and incompleteness of data.

Collected data can be in different formats, variety, and structure [3]. 3) Privacy of data in the context of big data [3]: There is public fear regarding the inappropriate use of personal data, particularly through linking of data from multiple sources. Managing privacy is both a technical and a Sociological problem. 4) Storage of huge amount of data [1] [3]: it represents the problem of how to recognize and store important information, extracted from unstructured data, efficiently. 5) Data visualization [1]: Data processing techniques

should be efficient enough to enable real time visualization.
6) Job scheduling in big data [4]: This problem focuses on efficient scheduling of jobs in a distributed environment. 7) Fault tolerance [5]: is another issue in Hadoop framework in big data. In Hadoop, NameNode is a single point of failure. Replication of block is one of the fault tolerance technique used by Hadoop. Fault tolerance techniques must be efficient enough to handle failure in distributed environment. MapReduce [6] provides an ideal framework for processing of such large datasets by using parallel and distributed programming approaches.

II. MAPREDUCE

MapReduce functionality depends on two functions: Map and Reduce function. Both functions are written by user. The Map function takes an input pair and generates a set of intermediate key/value pairs. The MapReduce library collect all intermediate values associated with the same intermediate key and transfers them to the Reduce function. The Reduce function gets an intermediate key with associated set of values. It combines these associated values to make a smaller set of values. The Figure 1 shows all process of



MapReduce.

Fig. 1The Overall MapReduce Word Count Process

III. HADOOP ARCHITECTURE

Scheduling decisions are taken by a master node, called the Job Tracker, and the worker nodes that called Task Tracker execute the tasks.

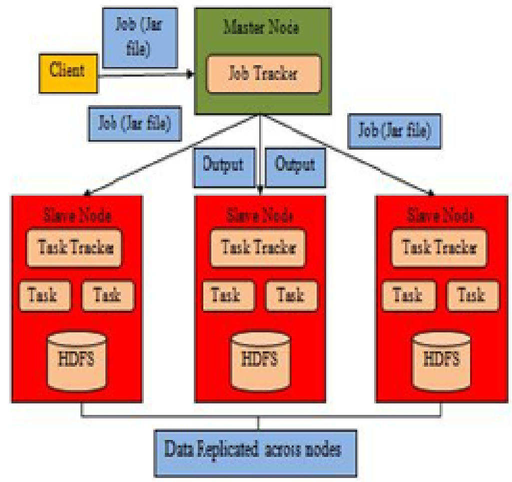


Fig. 2The Overall MapReduce Word Count Process

A Hadoop cluster includes a single master and multiple slave nodes. Figure 2 shows Hadoop Architecture. The single master node consists of a Job tracker, Task tracker, Name node and Data node.

A. Job Tracker

The primary function of the job tracker is managing the task trackers, tracking resource availability. The Job tracker is a node which controls the job execution process. Job tracker performs mapreduce tasks to specific nodes in the cluster. Client submits jobs to the Job tracker [51]. When the work is completed, the Job tracker updates its status. Client applications can ask the Job tracker for information.

B. Task Tracker

It follows the orders of the job tracker and updating the job tracker with its status periodically. Task tracker run tasks and send the reports to Job tracker, which keeps a complete record of each job. Every Task tracker is configured with a set of slots; it indicates the number of tasks that it can accept.

C. Name Node

The name node maps to, what block locations and which blocks are stored on which data node. Whenever a data node undergoes a disk corruption of a particular block, the first table gets updated and whenever a data node is detected to be dead due to network failure or a node, both the tables get updated. The updating of the table is based on only failure of the nodes. It does not depend on any neighbor blocks or any block locations to identify its destination. Each block is separated with its job nodes and respective allocated process.

D. Data Node

The node which stores the data in hadoop system is known to be as data node. All data nodes send a heartbeat message to the name node for every three seconds to say that they are alive. If the name node does not receive a heartbeat from a particular data node for ten minutes, then it considers that data node to be dead or out of service. It initiates some other data node for the process. The data nodes update the name node with the block information periodically [47].

IV. JOB SCHEDULING IN BIGDATA

The default Scheduling algorithm is supported on FIFO where jobs were executed in the magnitude of their humility. Later on the cognition to set the priority of a Job was added. Facebook and Character contributed meaningful apply in processing schedulers i.e. Legible Scheduler [8] and Capacity Scheduler [9] respectively which after free to Hadoop Dominion. This section describes various Job Scheduling algorithms in big data.

A. Default FIFO Scheduling

The default Hadoop scheduler operates using a FIFO queue. After a job is divided into independent tasks, they are ended into the queue and allotted to free slots as they get acquirable on Task Tracker nodes [50]. Although there is keep for decision of priorities to jobs, this is not revolved on by default. Typically apiece job would use the complete assemble, so jobs had to inactivity for their release. Regularize though a distributed constellate offers zealous latent for offering larger resources to numerous users, the job of intercourse resources even-handedly between users requires a turn scheduler [43]. Production jobs bet in a rational indication.

B. Fair Scheduling

The Fair Scheduler [8] was developed at Facebook to manage access to their Hadoop cluster and subsequently released to the Hadoop community. The Fair Scheduler plans to provide each user a fair share of the cluster capacity in excess of time. Users may allocate jobs to pools, with every pool owed a guaranteed smallest number of Map and Reduce slots. Free slots in unsuccessful pools may be owed to new pools; piece immoderateness ability within a pool is joint among jobs [45]. The Fair Scheduler maintains pre-emption, so if a pool has not received its fair contract for a destined period of measure, then the scheduler module veto tasks in pools flowing over capacity in dictate to afford the slots to the pool functional under capacity [46]. In addition, administrators may enforce priority settings on doomed pools. Tasks are therefore scheduled in an interleaved fashion, supported on their priority within their pool, and the constellate capacity and activity of their pool. As jobs contain their tasks assigned to Task Tracker slots for calculation, the scheduler follows the shortfall between the become of calculate really old and the saint fair percentage for that job. Eventually, this has the result of ensuring that jobs obtain roughly equal amounts of resources. Shorter jobs are assigned enough resources to terminate fast. Simultaneously, longer jobs are assured to not be ravenous of resources [44].

C. Capacity Scheduling

Capacity Scheduler [10] initially developed at Yahoo addresses a usage circumstances where the number of users is huge, and there is a require to make sure a fair assign of calculation resources between users.

The Capacity Scheduler allocates jobs supported on the submitting user to queues with configurable drawing of Map and Minify slots [11]. Queues that hold jobs are bestowed their organized capacity;

patch a trip capacity in a queue is shared among opposite queues. Within a queue, planning operates on a modified priority queue groundwork with specialized person limits, with priorities orientated supported on the quantify a job was submitted, and the priority scene allocated to that human and accumulation of job [22]. When a Task Tracker receptacle becomes unfixed, the queue with the lowest laden is elite, from which the oldest remaining job is chosen [25]. A task is then scheduled from that job [23]. This has the validity of enforcing meet capacity distribution among users, rather than among jobs, as was the case in the Fair Scheduler [24].

D. Dynamic Proportional Scheduling

According to Sandholm and Lai [12], Dynamic Proportional Scheduling provides more job sharing and prioritization that result in increasing share of cluster resources and more differentiation in service levels of different jobs [26]. This algorithm improves response time for multi-user Hadoop environments [27].

E. Resource-Aware Adaptive Scheduling (RAS)

To improve resource utilization across machines even as monitoring completion time, RAS proposed by Polo et al. [13] for Map Reduce with multi-job workloads.

Zhao et al. [14] gives task scheduling algorithm based on resource attribute selection (RAS) to determine its resource attributes by sending a set of test tasks to an execution node before a task is scheduled; and then select optimal node to execute a task according to resource requirements and appropriateness between the resource node and the task, which uses history task data if exists [30].

F. MapReduce task scheduling with deadline constraints (MTSD) algorithm

According to Tang et al. [15], scheduling algorithm sets two deadlines: map-deadline and reduce-deadline. Reduce-deadline is just the users' job deadline. Pop et al. [16] presents the classical approach for a periodic task scheduling by considering a scheduling system with different queues for periodic and a periodic tasks and deadline as the main constraint, and develops a technique to guess the number of resources required to schedule a set of an interrupted tasks, by considering together implementation and data transfers costs [33]. Based on a numerical model, and by using dissimilar simulation scenarios, MTSD proved the subsequent statements:

(1) Numerous sources of independent an episodic tasks can be measured approximating to a single one;

(2) when the number of estimated resources go beyond a data center capacity, the tasks migration between different regional centers is the suitable solution with respect to the global deadline; and (3) in a heterogeneous data center, we need higher number of resources for the same request with respect to the deadline constraints. In MapReduce, Wang and Li [17] elaborated the task scheduling, for disseminated data centers on heterogeneous networks through adaptive heartbeats, job deadlines and data locality [28]. Job deadlines are dividing along with the most data quantity of tasks. With the considered constraints, the task scheduling is twisted as an assignment problem in every heartbeat, in which adaptive heartbeats are intended by the processing times of tasks and jobs are sequencing in terms of the separated deadlines and tasks are planned by the Hungarian algorithm [29]. On the basis of data transfer and processing times, the most suitable data center for all mapped jobs are determined in the reduce phase [32].

G. Delay Scheduling

The objective is to address the conflict between locality and fairness [31]. When a node requests for a task, if the head-of-line job cannot project a local task, scheduler skip that task and looks at subsequent jobs. If a job has been skipped for long, we allow it to project non- local tasks, to avoid starvation [40]-[41].

Delay scheduling provisionally relaxes fairness to get better locality through allowing jobs to wait for scheduling on a node among local data. Song et al. [18] provide a game assumption based technique to solve scheduling issues by separating a Hadoop scheduling issue into two levels—job level and task level [42].

For the job level scheduling, use a bid model to provide guarantee to the fairness and reduce the average waiting time [34]. For tasks level, change scheduling problem into assignment problem and use Hungarian Method to optimize the problem. Wan et al. [19] gives multi-job scheduling algorithm in MapReduce based on game assumption which deals with the competition for resources between many jobs [35].

H. Multi Objective Scheduling

Nita et al. [20] give details about scheduling algorithm named MOMTH by considering objective functions associated to resources and users in thesimilar time with constraints similar to deadline and budget [48].

The implementation model consider as all MapReduce jobs are independent, there is no nodes failure before/during scheduling computation and scheduling decision is taken only based on the current knowledge. Bian et al. presents scheduling strategy based on fault tolerance [38]. According to this scheduling strategy, the cluster finds the speed of the current nodes and creates some backups of the intermediate MapReduce data results on to a high performance cache server [37]. The data produced by that node may go wrong soon. Thus the cluster may resume the execution to the previous level rapidly ifthere are several nodes going wrong, the Reduce nodes read the Map output from the cache server or from both the cache and the node, and keeps its high performance [49].

I. Hybrid Multistage Heuristic Scheduling (HMHS)

Chen et al. [21] elaborates heuristic scheduling algorithm named HMHS, which attempts to explain the scheduling trouble by splitting it into two sub problems: sequencing and dispatching. For sequencing, they make use of a heuristic based on Pri (the modified Johnson's algorithm) [36]. For dispatching, they suggest two heuristics Min-Min and Dynamic Min-Min.

TABLE I
COMPARISON OF VARIOUS JOB SCHEDULING ALGORITHMS IN BIGDATA

Scheduling Algorithm	**Technology**	**Advantages**	**Disadvantages**
Default FIFO Scheduling [22]	Schedule jobs based ontheir priorities in first-in first-out	Cost of entire cluster scheduling process is less. Simple to implementand efficient.	Designed only for singletype of job. Low performance whenrun multiple types of jobs. Poor response times forshort jobs compared to large jobs.
Fair Scheduling [8]	Do an equal distribution ofcompute resources amongthe users/jobs in thesystem.	Less complex Works well when bothsmall and large clusters. It can provide fast response times for smalljobs mixed with largerjobs.	Does not consider thejob weight of each node.
Capacity Scheduling [10]	Maximization the resourceutilization and throughputin multi-tenant clusterenvironment.	Ensure guaranteedaccess with the potential to reuse unused capacity andprioritize jobs withinqueues over large cluster.	The most complex Among three schedulers.
Dynamic Proportional Scheduling [12]	Designed for dataintensive workloads andtries to maintain data locality during job execution	It is a fast and flexiblescheduler. It improves responsetime for multi-user Hadoop environments.	If the system eventuallycrashes then all unfinishedlow priority processes getslost.
Resource-Aware Adaptive Scheduling (RAS) [13]	Dynamic Free SlotAdvertisement. Free SlotPriorities/Filtering	It improves the Jobperformance.	Only takes action onappropriate slow tasks.
MapReduce taskscheduling with deadline constraints (MTSD) [15]	Achieve nearly fulloverlap via the novel ideaof including reduce in theoverlap.	It Reduce computationtime. Improve performance For the important class of shuffle-heavy Map Reductions.	Better work with small clusters only.
Delay	To address the	Simplicity of	No particular

Scheduling [18]	conflictbetween locality andfairness.	scheduling	
Multi Objective Scheduling [20]	The implementation modelconsider as all MapReducejobs are independent, thereis no nodes failurebefore/during scheduling computation and scheduling decision is taken only based on thecurrent knowledge.	It keeps performance ishigh.	Execution Time is toolarge.
Hybrid Multistage Heuristic Scheduling (HMHS) [21]	Johnson's algorithm & Min-Min and Dynamic MinMin algorithm used	Achieves not only highdata locality rate but alsohigh cluster utilization.	It does not ensurereliability.

V. DISCUSSIONS

This paper provides the classification of Hadoop schedulers based on different parameters such as time, priority, resources etc. This paper discusses about how various task scheduling algorithms help in achieving better result in Hadoop cluster. Furthermore this paper also discusses about advantages and disadvantages of various task scheduling algorithms. This comparison results shows, each scheduling algorithm has some advantages and disadvantages. So, all algorithms are important in job scheduling.

VI. CONCLUSIONS

The paper gives an overall idea about different job scheduling in big data. It compares the properties of various task scheduling. Different scheduling techniques to enhance the data locality, make span, efficiency, fairness and performance are discussed in this paper. However, the scheduling technique is an open area for research.

REFERENCES

[1] Kousik, N. V., M. Sivaram, and S. Kalidass. "AUTONOMOUS GREEDY ROUTING IN WIRELESS SENSOR NETWORKS." ICTACT Journal on Communication Technology 10.1 (2019).
[2] Sivaram, Murugan et al. 'Data Fusion Using Tabu Crossover Genetic Algorithm in Information Retrieval'. 1 Jan. 2020 : 1 – 10.
[3] Prema, N.K., Sivaram, M. & Kousik, N.V. Divide and compress discrete cosine lossless compression coder to reduce dimensionality of test data. J Ambient Intell Human Comput(2020). https://doi.org/10.1007/s12652-020-02124-6.
[4] Raja, M. A., Ranathive, S., Sivaram, M., Krishna Kumar, L., Vinoth Kumar, K., & Amiri, I. (2019). Numerical Analysis of Soliton Propagation in a Tapered Waveguide, Journal of Optical Communications (published online ahead of print). doi: https://doi.org/10.1515/joc-2019-0118.
[5] Yuvaraj, D., et al. "An Efficient Lion Optimization Based Cluster Formation and Energy Management in WSN Based IoT." International Conference on Intelligent Computing & Optimization. Springer, Cham, 2019.
[6] Sivaram, M., et al. "Selection of Material using IWDNN Algorithm for Multiphase Material Component Design." 2019 Global Conference for Advancement in Technology (GCAT). IEEE, 2019.
[7] Sivaram, M., et al. "Biometric Security and Performance Metrics: FAR, FER, CER, FRR." 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). IEEE, 2019.
[8] Vadivel, Balachandran, M. Sivaram, and Ahmed Najat Ahmed. "English Language Significance from Ancient to the Modern-A Cram." 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). IEEE, 2019.
[9] Ranjeeth, S., et al. "Predicting Student Performance with ANNQ3H: A Case Study in Secondary Education." 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). IEEE, 2019.
[10] Rani, Shalli, et al. "A hybrid approach for the optimization of quality of service metrics of WSN." Wireless Networks 26.1 (2020): 621-638.
[11] Sivaram, M., et al. "Decision Support System for Determining Academic Advisor Using Simple Additive Weighting." Cognitive Informatics and Soft Computing. Springer, Singapore, 2020. 149-156.
[12] Sampathkumar, A., et al. "Advanced Energy Management System for Smart City Application Using the IoT." Internet of Things in Smart Technologies for Sustainable Urban Development. Springer, Cham, 2020. 185-194.
[13] Sivaram, M., et al. "A resilient and secure two-stage ITA and blockchain mechanism in mobile crowd sourcing." Journal of Ambient Intelligence and Humanized Computing (2020): 1-14.
[14] Kuppusamy, P. G., et al. "A Novel Approach Based on Modified Cycle Generative Adversarial Networks for Image Steganography." Scalable Computing: Practice and Experience 21.1 (2020): 63-72.
[15] Samikannu, Ravi. "An Efficient Image Analysis Framework for the Classification of Glioma Brain Images using CNN Approach." Computers, Materials & Continua 63.3 (2020): 1133-1142.
[16] El-Hageen, H. M., Kuppusamy, P.G., Alatwi, A. M., Sivaram, M., Yasar, Z., & Zaki Rashed, A. (2020). Different modulation schemes for direct and external modulators based on various laser sources, Journal of Optical Communications (published online ahead of print), 000010151520200029. doi: https://doi.org/10.1515/joc-2020-0029.
[17] Shankar, A., Sivakumar, N.R., Sivaram, M. et al. Increasing fault tolerance ability and network lifetime with clustered pollination in wireless sensor networks. J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-02325-z.
[18] Yuvaraj, D., M. Sivaram, and S. Navaneetha Krishnan. "Intelligent detection of untrusted data transmission to optimize energy in sensor networks." Journal of Information and Optimization Sciences (2019): 1-13.
[19] Gomathy, V., et al. "Malicious node detection using heterogeneous cluster based secure routing protocol (HCBS) in wireless adhoc sensor networks." Journal of Ambient Intelligence and Humanized Computing (2020): 1-7.
[20] Sivaram, M., et al. "Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud." JOURNAL OF AMBIENT INTELLIGENCE AND HUMANIZED COMPUTING(2020).
[21] Sampathkumar, A., Jaison Mulerikkal, and M. Sivaram. "Glowworm swarm optimization for effectual load balancing and routing strategies in wireless sensor networks." Wireless Networks (2020): 1-12.
[22] Maseleno, Andino, et al. "School now, pay later." Revista San Gregorio 1.32 (2019): 196-203.

[23] Yuvaraj, D., et al. "Some Investigation on DDOS Attack Models in Mobile Networks." (2019): 71-88.
[24] Sampathkumar, A., et al. "An efficient hybrid methodology for detection of cancer-causing gene using CSC for micro array data." Journal of Ambient Intelligence and Humanized Computing (2020): 1-9.
[25] Samanta, Debabrata, et al. "Distributed Feedback Laser (DFB) for Signal Power Amplitude Level Improvement in Long Spectral Band." Journal of Optical Communications 1.ahead-of-print (2020).
[26] Dhivakar, B., S. V. Saravanan, and R. Abirama Krishnan. "Statistical Score Calculation of Information Retrieval Systems using Data Fusion Technique." Computer Science and Engineering 2.5 (2012): 43-45.
[27] Sivaram, M., B. DurgaDevi, and J. Anne Steffi. "Steganography of two LSB bits." International Journal of Communications and Engineering 1.1 (2012): 2231-2307.
[28] Nithya, S., et al. "Intelligent based IoT smart city on traffic control system using raspberry Pi and robust waste management." Journal of Advanced Research in Dynamical and Control Systems, Pages (2018): 765-770.
[29] Abraham, Ste□n, Tana Luciya Joji, and D. Yuvaraj. "Enhancing vehicle safety with drowsiness detection and collision avoidance." International Journal of Pure and Applied Mathematics 120.6 (2018): 2295-2310.
[30] Batri, K., and M. Sivaram. "Testing the impact of odd and even point crossover of genetic algorithm over the data fusion in information retrieval." European Journal of Scientific Research(2012).
[31] Sivaram, M. "Odd and even point crossover based Tabu ga for data fusion in Information retrieval." (2014).
[32] Sivaram, M., et al. "Emergent news event detection from facebook using clustering." Journal of Advanced Research in Dynamical and Control Systems, Pages 1947 (1941): 2018.
[33] Malathi, N., and M. Sivaram. "An Enhanced Scheme to Pinpoint Malicious Behavior of Nodes In Manet's." (2015): 303-310.
[34] Punidha, R. "Integer wavelet transform based approach for high robustness of audio signal transmission." International Journal of Pure and Applied Mathematics 116.23 (2017): 295-304.
[35] Mahalakshmi, K. and M, Sivaram and Kumari, K. Shantha and Yuvaraj, D and Keerthika, R., Healthcare Visible Light Communication (2018). International Journal of Pure and Applied Mathematics, Volume 118 No. 11 2018, 345-348, Available at SSRN: https://ssrn.com/abstract=3352115.
[36] Punidha, R., K. Pavithra, and R. Swathika. "Preserving DDoS Attacks Using Node Blocking Algorithm." International Journal of Pure and Applied Mathematics 119.15 (2018): 633-640.
[37] Sivaram, M., et al. "Retransmission DBTMA protocol with fast retransmission strategy to improve the performance of MANETs." IEEE Access 7 (2019): 85098-85109.
[38] Sivaram, M., et al. "An Optimal Least Square Support Vector Machine Based Earnings Prediction of Blockchain Financial Products." IEEE Access 8 (2020): 120321-120330.
[39] Sivaram, M., et al. "Expert System in Determining the Quality of Superior Gourami Seed Using Forward Chaining-Based Websites." International Conference on Emerging Technologies in Computer Engineering. Springer, Singapore, 2019.
[40] Sivakumar, P., et al. "A novel free space communication system using nonlinear InGaAsP microsystem resonators for enabling power-control toward smart cities." Wireless Networks 26.4 (2020): 2317-2328.
[41] Yuvaraj, D., et al. "Shape, color and texture based CBIR system using fuzzy logic classifier." CMC-COMPUTERS MATERIALS & CONTINUA 59.3 (2019): 729-739.
[42] Shobana, S. Jeya, et al. "FCCP–NS: A Fair Congestion Control Protocol with N-Sinks in Wireless Sensor Networks." International Journal of Advanced Trends in Computer Science and Engineering 8.1.2 (2019).
[43] Venkatachalam, K., et al. "A novel method of motor imagery classification using eeg signal." Artificial Intelligence in Medicine 103 (2020): 101787.
[44] Rathee, Geetanjali, et al. "A trust computed framework for IoT devices and fog computing environment." Wireless Networks26.4 (2020): 2339-2351.
[45] Sivaram, M., et al. "The Real Problem Through a Selection Making an Algorithm that Minimizes the Computational Complexity." International Journal of Engineering and Advanced Technology 8.2 (2018): 95-100.
[46] Sivaram, M., et al. "DETECTION OF ACCURATE FACIAL DETECTION USING HYBRID DEEP CONVOLUTIONAL RECURRENT NEURAL NETWORK." ICTACT Journal on Soft Computing 9.2 (2019).
[47] Sivaram, M., et al. "Saving Private Money." Revista San Gregorio 32 (2019): 204-211.
[48] Sivaram, M., et al. "Improved Enhanced Dbtma with Contention-Aware Admission Control to Improve the Network Performance in Manets." CMC-COMPUTERS MATERIALS & CONTINUA 60.2 (2019): 435-454.
[49] Manikandan, V., et al. "Nature Inspired Improved Firefly Algorithm for Node Clustering in WSNs." CMC-COMPUTERS MATERIALS & CONTINUA 64.2 (2020): 753-776.
[50] Manikandan, V., and N. Yuvaraj. "Advanced Expert System Using Particle Swarm Optimization Based Adaptive Network Based Fuzzy Inference System to Diagnose the Physical Constitution of Human Body." Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics: Second International Conference, ICETCE 2019, Jaipur, India, February 1–2, 2019, Revised Selected Papers. Vol. 985. Springer, 2019.
[51] Divya, P., and V. Manikandan. "International Journal of Innovative Research in Engineering Science and Technology http://journal. selvamtech. com AN ENERGY SAVING TECHNIQUE FOR WIRELESS SENSOR NETWORKS."